Quorum Set Systems for Generalized Resource Allocation Problem

DWILYA MAKIWAN[1)] ARMIN LAWI[2)]

Abstract: Mutual exclusion (mutex) is fundamental problems in distributed systems in which a resource must be used at most one process at a time when some processes were communicating each other to compete in accessing. Many resource allocations problems have been constructed base on the mutual exclusion problem, and it can be generally formed to problems called (n,m,k,d)-resource allocation and further extended to (n,m,k,d_i)-resource allocation, i.e, the problem to synchronize accesses on k identical resources among n processes which belong to m groups and each resource can be used by at most d_i processes of the same group at a time. The generalized resource allocation problem can be solved using a quorum system called (m,k,d)-coterie and (m,k,d_i)-coterie that satisfies the relaxed safety property of the original mutual exclusion. If quorum based algorithm of mutual exclusion directly adopt to the quorum systems then the massage complexity is the same and the maximum degree of concurrency of the generalized resource allocation problem of $\sum_{i=1}^{h} d_i$ is achieved.

Keywords: Mutual exclusion, quorum systems, resources allocation, coteries.

1. Introduction

Algorithm for the mutual exclusion problem must dealing with messages delay; therefore it needs algorithm based on set systems called quorum-based algorithm. The synchronization algorithm based on quorum system is a robust approach since it is significantly tolerated the failures of nodes and communications that can lead to network partitioning. Every two quorums must satisfy intersection property and each node has only one permission to issue. In a quorum system, if a process request and receives permission from all nodes in a selected set, then it can enter the critical section. Therefore, it guarantees the safety condition in which at most one process is on critical section at a time. It must guarantee the satisfying of safety conditions, i.e., at most one process accessing resource at the same time, and liveness properties, i.e., a process requesting a resource will eventually succeed [2,4,6,7].

There have been proposed a new solution for h-out of k-Mutual Exclusion and it gave us a new comprehension about a quorum system called (m,h,k)-coterie [7],and further it have been expanded to solving (n,m,k,d)-resource allocation problem [4]. Basically, this paper presents the construction of (m,k,d)-coterie and (m,k,d_i)-coterie that has been studied in [3,6,7] with some addition to make the construction more robust and satisfy the required properties of the generalized resource allocation problem which is the safety property is relaxed from the original mutual exclusion problem.

2. Related Work

Mutual exclusion aims to synchronizing all the access of processes in distributed systems into single indivisible resource [9]. One of algorithm that can solve the problem of processes communication is algorithm introduced by Lamport **[5]**, when a process wants to access resources, it has to had permission from another process on the system by comparing smallest time stamp [9]. As follow to ensure safety, for mutual exclusion every two quorum (subsets of processes collection) must intersect, this helps increase fault tolerance because if a quorum is hit due to that some member in the quorum has failed, some other quorum may still be available to grant access of resource. A simple approach is processes compete for quorums as usual, but only one quorum can be acquired at a time. After a process has successfully acquired a quorum, it “helps” other processes of the same group to enter the critical section directly without being blocked by the quorums they are waiting for [3].

3. Set Systems for Resource Allocation Problem

To learn about quorum set systems, give a basic definition of coterie.

Definition 1 *Let P be a finite set of elements where* $C \subseteq 2^P$ *is nonempty set of P, called coterie under P if and only if:*
Minimality*:* $(\forall\ Q,Q' \in C)\ [Q \not\subset Q']$. (Q,Q’∈C is **quorum**)
Intersection*:* $(\forall\ Q,Q' \in C)[Q \cap Q' \neq \emptyset]$.

Example 1
Let $C_1 = \{\{1,2\}, \{2,3\}\}$, $C_2 = \{\{2,4\}, \{3,4\}, \{1,4\}$ coterie under $P = \{1,2,3,4\}$. So that all {1,2}, {2,3}, {2,4}, {3,4} and {1,4} are quorums.

For every set system is said to be solution of resource allocation problem if the following conditions hold:

Safety : There is only one process can access the resource at any time.

Liveness : a process requesting a resource will eventually succeed.

3.1 Quorum system for (*n,m,k,d*)-resources allocation

(n,m,k,d)-resources allocation problem concerns the scheduling of k identical resources among n processes which belong to m groups. Each resource can be used by at most d processes of the same time.

3.1.1 (*m,k,d*)-coterie

Definition 2 *(m,k,d)-coterie) A collection of sets* $\mathbf{M}=(C_1, \ldots, C_m)$

[1)] Department of Mathematics, Faculty of Mathematics and Natural Sciences, Hasanuddin University, Indonesia..
[2)] Faculty of Mathematics and Natural Sciences, Hasanuddin University, Indonesia.

where C_i is a d-coterie under P, $\forall C_i \in \boldsymbol{M}$ is a (m,k,d)-coterie if following condition holds:

Disjoint: *$\forall L = (C_1', \ldots, C_\ell') \in \boldsymbol{M}$ for any $\ell < m$ there exists element $C \in \boldsymbol{M}$ such that C and C_i are disjoint $\forall 1 \le i \le l$*

Bicoterie: *$\forall K = (C_1', \ldots, C_k') \subseteq \boldsymbol{M}$, there exists pair a (C_i', C_j') forms bicoterie $\forall 1 \le i \ne j \le k$.*

3.1.2 Construction of (*m*,*k*,*d*)-coterie

Assumed $n = 2kd^2$, $m = 2k$, partition $\mathcal{P}$ into m subsets of $P_1, \ldots, P_m$. Create *d-coterie* C_i on each set P_i by construct *d* disjoint sets (quorums) Q_{ij}, $1 \le j \le d$, $1 \le i \le m$.

Hence,

$$\mathcal{P} = \cup_u P_u \text{ and } P_u \cap P_v = \emptyset, \forall\, 1 \le u, v \le \text{m} \quad (1)$$

$$Q_{ij} = \{p_{js}^i \in P_i, 1 \le s \le d\} \quad (2)$$

$$Q_{(i+1)j} = \{p_{js}^{i+1} \in P_{i+1} 1 \le s \le d\} \quad (3)$$

$$C_i = Q_{ij}^* = Q_{ij} \cup Q_{(i+1)j} \bmod^{a} m$$

$$= Q_{ij} \cup (\cup p_{js}^{i+1} \in P_{i+1}) \bmod m \quad (4)$$

with, $|Q_{ij}^*| = 2d$ if $|Q_{ij}| = d$,

$$|Q_{ij}^*| = 2d + 1 \text{ if } |Q_{ij}| = d + 1,$$

$$|Q_{ij}^*| = 2d + n \text{ if } |Q_{ij}| = d + n.$$

Let $\boldsymbol{M} = (C_1, \ldots, C_m)$ with $C_i = \{Q_{i1}, \ldots, Q_{id}\}$.

Lemma 1. C_i is *d-coterie* under $(P_i \cup P_{i+1})$ for $1 \le i \le m$.

Proof. $P_1, \ldots, P_m \in \mathcal{P}$ with condition of (1), $C_i \in P_i$ and $C_{i+1} \in P_{i+1}$. It will shows C_i is *d-coterie* under $(P_i \cup P_{i+1})$

As we know $C_i \in P_i$, $C_{i+1} \in P_{i+1}$, and (2), (3) represent

$$Q_{ij} = \{p_{js}^i \in P_i, 1 \le s \le d\}$$

$$Q_{(i+1)j} = \{p_{js}^{i+1} \in P_{i+1}, 1 \le s \le d\}$$

So that, $C_i = Q_{ij}$ and $C_{i+1} = Q_{(i+1)j}$ with $Q_{ij}, Q_{(i+1)j}$ are disjoint sets and because from (4) $C_i = Q_{ij}^* = Q_{ij} \cup Q_{(i+1)j} \, mod\, m$ we can see C_i is a disjoint sets with $Q_{ij}^* \in P_i \cup P_{i+1}$. ∎

Lemma 2. (C_i, C_{i+1}) is a bicoterie $\forall 1 \le i \le m - 1$ and (C_i, C_{i+2}) is a disjoint $\forall 1 \le i \le m - 2$.

Proof. By partitioning $\mathcal{P}$ into $P_1, \ldots, P_m$, and using (1) appear collections $C_i \in P_i$, $C_{i+1} \in P_{i+1}$, $C_{i+2} \in P_{i+2}$. It shows (C_i, C_{i+1}) is a bicoterie and (C_i, C_{i+2}) is a disjoint.

From (2) and (3) we directly construct

$$C_i = Q_{ij} = \{p_{js}^i \in P_i, \quad 1 \le s \le d\}$$

$$C_{i+1} = Q_{(i+1)j} = \{p_{js}^{i+1} \in P_{i+1} 1 \le s \le d\}$$

$$C_{i+2} = Q_{(i+2)j} = \{p_{js}^{i+2} \in P_{i+2} 1 \le s \le d\}$$

By joining these elements according to construction on (4) shows

$$C_i = Q_{ij}^* = Q_{ij} \cup Q_{(i+1)j} \bmod m,$$

$$C_{i+1} = Q_{(i+1)j}^* = Q_{(i+1)j} \cup Q_{(i+2)j} \bmod m,$$

$$C_{i+2} = Q_{(i+2)j}^* = Q_{(i+2)j} \cup Q_{(i+3)j} \bmod m.$$ ∎

Theorem 1. *M* is $\left(m, \frac{m}{2}, \sqrt{\frac{n}{m}}\right)$-coterie, with size of each coterie is $\left(d = \sqrt{\frac{n}{m}}\right)$ the size of each quorum is $\left(Q_{ij} = 2\sqrt{\frac{n}{m}}\right)$. ∎

For more general with $m = tk, k = \frac{m}{t}$, and $t = 1,2,3, \ldots, \left\lfloor \frac{m}{k} \right\rfloor$, construction can be done by a similar procedure. Partition set $\mathcal{P}$ into P_i, and create *d-coterie* by create *d* disjoint sets or quorum

$$C_i = \{Q_{ij}\} \text{ of } P_i . 1 \le j \le d, 1 \le i \le m \quad (5)$$

$$Q_{ij} = \{p_{js}^i \in P_i, 1 \le s \le k\} \quad (6)$$

$$Q_{(i+1)j} = \{p_{js}^{i+1} \in P_{i+1} 1 \le s \le k\} \quad (7)$$

Let coterie $C_i = Q_{ij}^* = Q_{ij} \cup Q_{(i+1)j} \bmod m$, then

$$Q_{ij}^* = Q_{ij} \cup (\cup_{1 \le v \le t-1} \{p_{js}^{i+v} \in P_{i+v}, 1 \le s \le d\}) \ldots (8)$$

1. $|Q_{ij}^*| = td$ if $|Q_{ij}| = d$, $|Q_{ij}^*| = td + 1$ if $|Q_{ij}| = d + 1$,

   and $|Q_{ij}^*| = td + n$ if $|Q_{ij}| = d + n$.
2. $Q_{ij}^* \cap Q_{(i+t)j'}^* = \emptyset$, $1 \le j, j' \le d$
3. $Q_{ij}^* \cap Q_{(i+v)j'}^* \ne \emptyset$, $1 \le v < t$ and $1 \le j, j' \le d$

By the definition, $M = (C_1, \ldots, C_m)$ is coterie.

Theorem 2. *M* is a $\left(m, \frac{m}{t}, \sqrt{\frac{n}{m}}\right)$-coterie with $t = 1,2,3, \ldots, \left\lfloor \frac{m}{k} \right\rfloor$.

Proof. $M = (C_1, \ldots, C_m)$ and it shows *M* is $\left(m, \frac{m}{t}, \sqrt{\frac{n}{m}}\right)$–coterie under $\mathcal{P}$. $M = (C_1, \ldots, C_m)$ where $C_i = Q_{ij}^*$ with $C_i = Q_{ij}^* = Q_{ij} \cup Q_{(i+1)j} \bmod m$. By using the result on (2) and (3), $Q_{ij}^* \cap Q_{(i+t)j'}^* = \emptyset$ and $Q_{ij}^* \cap Q_{(i+v)j'}^* \ne \emptyset$ means satisfying of disjoint and bicoterie properties, and finally we can say that $\boldsymbol{M}$ is a $\left(m, \frac{m}{t}, \sqrt{\frac{n}{m}}\right)$-coterie. ∎

Example 2.

The quorum system $M = (C_1, C_2, C_3, C_4)$ is (4,2,2)-coterie under $P = \{1, \ldots, 16\}$. Assumed $n = 2kd^2$ will sets up into each of sets of coterie. Insert element Q_j on C_{i+1} into C_i thus we have

$C_1 = \{\{1,2,5,7\}, \{3,4,6,8\}\}$,
$C_2 = \{\{5,6,9,11\},\{7,8,10,12\}\}$,
$C_3 = \{\{9,10,13,15\},\{11,12,14,16\}\}$,
$C_4 = \{\{1,3,13,14\},\{2,4,15,16\}\}$.

Each pair C_1 and C_3, C_2 and C_4 are *disjoint* because all element on each pair set was not same. But for pair sets of C_1 and C_3, C_2 and C_4 are *bicoterie* because each element was intersected with their pair sets.

[a] all indexes i and j working on modulo m

3.2 Quorum system for (n,m,k,d_i)-resource allocation

For resource allocation problem with at most d_i (different degree) processes at the same time, it will solved by using quorum system called (m,k,d_i)-coterie.

3.2.1 (m,k,d_i)-coterie.

Definition 3 *(m,k,d_i)-coterie). A collection set* $\mathbf{M} = (C_1, ..., C_m)$*. And* C_i *is* d_i*-coterie under P, for every* $1 \leq d_i < n$*, called (m,k,d_i)-coterie under P if disjoint and bicoterie hold.*

By using this definition, we definitely can constructed some steps to built (m,k,d_i)-coterie to solve the (n,m,k,d_i)-resource allocation.

3.2.2 Construction of (m,k,d_i)-coterie

(m,k,d_i)-coterie can be constructed by using the modification of (m,k,d)-coterie by assuming $m = tk$ and $n = tk\sum_{i=1}^{m} d_i\, d_{i-1} \pmod m$. We first partition $\mathcal{P}$ into P_i to create coterie d_i, where $d_i \subseteq Q_{ij}$, $C_i = \{Q_{ij}\}$ ………..…… (9)

$$Q_{(i+1)j} = \{p_{js}^{i+1} \in P_{i+1}\,, 1 \leq s \leq d\} \text{………… (10)}$$

with $|Q_{(i+1)j}| = d_{Q_{ij}}$

$$C_i = Q_{ij}^* = Q_{ij} \cup Q_{(i+1)j} \bmod m$$

$$= Q_{ij} \cup \left(\cup_{1 \leq v \leq t-1}\{p_{js}^{i+v} \in P_{i+v}\,, 1 \leq s \leq d\} \bmod m\right) \text{… (11)}$$

Finally, we have

$$Q_{ij}^* \cap Q_{(i+t)j'}^* = \emptyset \;,\; 1 \leq j,\; j' \leq d_i$$

$$Q_{ij}^* \cap Q_{(i+v)j'}^* \neq \emptyset, 1 \leq v < t \;\text{ and }\; 1 \leq j, j' \leq d_i$$

And $M = (C_1, ..., C_m)$ is coterie

Theorem 3 M is$\left(m, \frac{m}{t}, d_i\right)$-coterie with $t = 1,2,3, ..., \left\lfloor \frac{m}{k} \right\rfloor$.

Proof. Let $M = (C_1, ..., C_m)$.It shows M is $\left(m, \frac{m}{t}, d_i\right)$-coterie under $\mathcal{P}$. By using the construction (11)

$$C_i = Q_{ij}^* = Q_{ij} \cup Q_{(i+1)j} \;\; mod\; m \qquad ,$$

$$Q_{ij}^* = Q_{ij} \cup \left(\cup_{1 \leq v \leq t-1}\{p_{js}^{i+v} \in P_{i+v}\,, 1 \leq s \leq d\} \bmod m\right)$$

bringing us into result on (4) and (5) with satisfying of disjoint for $Q_{ij}^* \cap Q_{(i+t)j'}^* = \emptyset$ and bicoterie for $Q_{ij}^* \cap Q_{(i+v)j'}^* \neq \emptyset$

and it directly shows M is $\left(m, \frac{m}{t}, d_i\right)$-coterie. ■

Example 3.
With $m = 2k$, $m = 4$, $k = 2$ and $d = \{2,3,3,2\}$ we will construct (m,k,d_i)-coterie. Partition $\mathcal{P}$ into 10 subsets each quorum. Sets it up into 4 different coteries
C_1 = {{1,2,5,7,9},{3,4,6,8,10}}
C_2 = {{5,6,11,14,17},{7,8,12,15,18}{9,10,13,16,19}}
C_3 ={{11,12,13,20,23},{14,15,16,21,24},{17,18,19,22,25}}
C_4 = {{1,3,20,21,22},{2,4,23,24,25}}
Each pair C_1 and C_3, C_2 and C_4 are *disjoint* because all element on each pair set was not same. But for pair sets of C_1 and C_3, C_2 and C_4 are *bicoterie* because each element was intersected with their pair sets.
Now let more understand about the communication using the (m,k,d)-coterie by having the algorithm below.

Trying Section { *//When* p_i *wishes to access a resource* r_v
1: ts_i++; // ts_i *is* p_i *is current logical time*
2: *Select a quorum Q in* C_v;
3: **send** req(ts_i, p_i) **to** p_j , $\forall p_j \in Q$;
4: *Add* p_j ($\in Q$) *answering* ack *into* AGREE$_i$;
5: **if** (*there is a* $Q(\in C_v) \subseteq$ AGREE$_i$) {
6: *state* := **Critical Section**; }
7: **else-if** { // *If there exists* p_j ($\in Q$) *answers* wait
8: *Add* p_j *answering* wait *into* DISAGREE$_i$;
9: *Select another quorum* $Q' \in C_v$ *such that*
$Q' \cap$ DISAGREE$_i = \emptyset \wedge Q = \max\{|Q \cap \text{AGREE}_i|\}$;
10: **if** (*there is no quorum satisfy*) {
11: *state* := **Wait**; }
12: $Q := (Q' - Q)$ *and* **go to** line 3; } }

Exit Section *{ // When user* p_i *leaves resource* r_v *}*
1: **send** exit **to** $\forall p_j \in$ (AGREE$_i$ ∪ DISAGREE$_i$);

When p_i **receives** req(ts_j, p_j) **message:**
1: **if** (PERM$_i = \emptyset$) {
2: **send** ack **to** p_j and *add* req(ts_j, p_j) *to* PERM$_i$; *}*
3: **else-if** { // If there exists (ts_x, p_x) $\in$ PERM$_i$
4: // Let (ts_y, p_y) is the highest priority in QUEUE$_i$;
5: *Insert* req(ts_j, p_j) *into* QUEUE$_i$;
6: **if** ((ts_j, p_j) > max*{*(ts_x, p_x), (ts_y, p_y)*}*) {
7: **send** reclaim **to** p_x; }
8: **else-if** {
9: **send** wait **to** p_j ; }}

When p_i **receives** exit **message from** p_j **:**
1: *Remove request* p_j *from* PERM$_i$;
2: **if** (QUEUE$_i \neq \emptyset$) {
3: // Let (ts_y, p_y) is the highest priority in QUEUE$_i$;
4: *Move* req(ts_y, p_y) *from* QUEUE$_i$ *to* PERM$_i$;
5: **send** ack **to** p_y; }

When p_i **receives** reclaim **message from** p_j **:**
1: **if** (p_i *not in critical section* and $p_j \in$ AGREE$_i$) {
2: *Move* p_j *from* AGREE$_i$ *to* DISAGREE$_j$;
3: **send** relinquish **to** p_j ; }

When p_i **receives** relinquish **message from** p_j **:**
1: // Let (ts_x, p_x) is the highest priority in QUEUE$_i$;
2: **if** (ts_x, p_x) > (ts_j, p_j) {
3: *Move* req(ts_j, p_j) *from* PERM$_i$ *to* QUEUE$_i$;
4: **send** ack **to** p_x;
5: *Move* req(ts_x, p_x) *from* QUEUE$_i$ *to* PERM$_i$; }

Fig. 1. The (m, k, d)-coterie based algorithm

Theorem 4. *The algorithm in Fig. 1 adopting* (m, h, k_i)*-coteries achieves the maximum degree of concurrency*

The (n,m,k,d)-resource allocation problem can solved using quorum system called (m,k,d)-coterie since d-mutual exclusion with d-access together satisfying of disjoint and bicoterie conditions. This problem also free from *deadlock* and *starvation* because the execution of resources at the end will be succeed since it adopts Lamport algorithm directly with timestamp method (ts_i, p_i). The same stands for (n,m,k,d_i)-resources allocation with quorum system called (m,k,d_i)-coterie**.**

4. Conclusion

A (m,k,d)-coterie by partitioning universe set into some disjoint d-coterie, may be one way to resolve (n,m,k,d)-resource allocation problem with each size of new quorum is $|Q^*_{ij}| = td$ and its depend on the number of elements that inserted from previous quorum $Q_{(i+1)j}$. The construction of (m,k,d_i)-coterie leads us to solution of (n,m,k,d_i)-resource allocation with capacity of d_i resources is different, both of it guarantee *safety* and *liveness* by using timestamp algorithm. If *mutex* system straight adopted into quorum system that is (m,k,d_i)-coterie, then the massages complexity is $\sum_{i=1}^{h} d_i$.

References

[1] Garcia-Molina, H., Barbara, D. 1985. How to assign votes in a distributed system. J . ACM 32(4), 841–860.

[2] Harada Takashi, Yamashita Masafumi. 2001. Coterie Join Operation and Tree Structured k-Coteries, IEEE Transaction on Parallel and Distributed Systems, Vol12, Number 9.

[3] Joung, Yuh-Jzer. 2010. On quorum sistem for group resources allocation, Distributed Computing, Vol 22, number 3, 197214, DOI10.1007/s00446-010-0094-4.

[4] Jiang. 2005. Quorum Systems. http://in1.csie.ncu.edu.tw/ ~jrjiang/alg2005/QuorumSystems.ppt.

[5] Kshemkalyani Ajay, Singhal Mukesh. 2008. *Distributed Computing Principles, Algorithm and Sistem*. Cambridge University Press.

[6] Lawi, A., Oda, K., dan Yoshida, T. 2006. Quorum based distributed conflict resolution algorithm for bounded capacity resources, Lecture Notes in Computer Science (LNCS), Vol. 4331, 135–144..

[7] Lawi, A., Yamashita, M.: A quorum based *m*-group (*h, k*)-exclusion algorithm. Proc. International Symposium on Information Science and Electrical Engineering (ISEE2003). 405–408

[8] Najafi, S. 2008. A New GA - *Based and Graph Theory Supported Distribution System Planning*, Vol. 2, No. 8. ISSN – 1453 – 1119.

[9] Tanenbaum, A, and Steen, M.V. Oct. 2006. *Distributed Systems: Principles and Paradigms*,Second Edition. Prentice Hall.

[10] Tel, G. October 2000. *Introduction to Distributed Algorithms*. Cambridge University Press.